

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first module through an interface with capabilities shared by all the interacting modules, wherein the modules are stored in computer storage, the method comprising the computer-implemented steps of:

creating [[and storing]] first information describing the computer program module interface capabilities at [[a plurality of]] one or more times [[with]] ;

storing the first information in a corresponding plurality of instances of a data structure wherein each instance of the data structure corresponds to the interface capabilities at one or more plurality of times;

creating and storing a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first computer program module;

automatically [[developing]] assigning a second version number for a second computer program module of the one or more interacting modules based on a corresponding instance from the plurality of instances of the data structure [[and]] contained in the mapping and based on second information describing interface capabilities of the second computer program module;

determining compatibility of the modules based on a first version number for the first module and the second version number for the second module.

2. (Currently Amended) A method as recited in Claim 1, wherein the step of automatically [[developing]] assigning comprises:

describing a subset of the interface capabilities, which subset is employed by the second computer program module;

determining [[of]] from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and

assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

3. (Original) A method as recited in Claim 2, wherein said step of assigning the second version number comprises assigning as the second version number a particular value of the plurality of version numbers for the first module, the particular value associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

4. (Currently Amended) A method as recited in Claim 1, wherein the plurality of version numbers for the first computer program module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

5. (Currently Amended) A method as recited in Claim 1, wherein said step of describing the computer program module interface capabilities includes generating and storing in a first instance of the data structure data indicating signatures of a plurality of routines of the interface at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

6. (Original) A method as recited in Claim 1, wherein:
each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;
the first computer program module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

7. (Currently Amended) A method as recited in Claim 1, wherein the second version number is [[developed]] assigned when the second module is developed; and compatibility is determined at a later time.

8. (Currently Amended) A method of maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the method comprising the steps of:

retrieving data from a stored mapping between a plurality of instances of a data structure describing capabilities of the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module;

automatically [[developing]] assigning a second version number for a second module of the one or more interacting modules based on the mapping and based on a description of interface capabilities of the second module;

determining compatibility based on a first version number for the first module and the second version number for the second module.

9. (Currently Amended) A method as recited in Claim 8, said step of assigning [[developing]] the second version number further comprising the steps of:

describing a subset of the interface, which subset is employed by the second module;

determining of the plurality of instances at least one instance including data describing the subset of the interface; and

assigning the second version number for the second module based on the mapping and the at least one instance.

10. (Currently Amended) A method as recited in Claim 9, wherein said step of assigning the second version number comprises assigning as the second version number a particular value of the plurality of version numbers for the first computer program module, the particular value

associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

11. (Currently Amended) A method as recited in Claim 8, wherein the plurality of version numbers for the first computer program module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

12. (Original) A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating a routine name and a routine type of a routine of the first module at a first time.

13. (Original) A method as recited in Claim 12, wherein the first data further indicates a parameter type for the routine.

14. (Currently Amended) A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating signatures of a plurality of routines of the computer program module interface capabilities at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

15. (Currently Amended) A method as recited in Claim 14, wherein the data indicating signatures of the plurality of routines of the computer program interface capabilities at a first time comprises hashed values, each hashed value uniquely indicating a signature of each routine of the interface.

16. (Original) A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface.

17. (Original) A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface except routines not implemented in the first module.

18. (Currently Amended) A method as recited in Claim 8, wherein:
the second version number is developed when the second computer program module is
developed; and
compatibility is determined at a later time.
19. (Currently Amended) A method as recited in Claim 18, wherein compatibility is
determined when the second computer program module is installed for use with the first
computer program module.
20. (Currently Amended) A method as recited in Claim 18, wherein compatibility is
determined when the second computer program module is invoked for execution by the first
module.
21. (Currently Amended) A method as recited in Claim 8, wherein each computer program
module of the first computer program module and the one or more interacting computer program
modules comprises instructions for causing one or more processors to perform one or more tasks.
22. (Currently Amended) A method as recited in Claim 8, wherein the first computer program
module comprises instructions for causing one or more processors to manage a plurality of
networking [[devices]] devices in response to data indicating input by a user.
23. (Currently Amended) A method as recited in Claim 8, wherein each interacting computer
program module of the one or more interacting computer program modules comprises
instructions for causing one or more processors to provide device-specific information for one of
a plurality of networking [[devices]] devices.
24. (Currently Amended) A method as recited in Claim 21, wherein:
each of the interacting computer program modules include instructions causing one or
more processors to obtain at least one of a property of a corresponding networking

device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

25. (Currently Amended) A method of determining version compatibility between a first computer program module and a computer program second module of one or more interacting modules that interact with the first computer program module through an interface, the method comprising the steps of:

obtaining a first version number for the first computer program module;

obtaining a second version number for the second computer program module, the second version number set when the second computer program module is developed based on a mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module and based on a description of interface capabilities of the second computer program module;

and

determining whether the computer program modules are compatible based on the first version number and the second version number.

26. (Currently Amended) A method as recited in Claim 25, wherein:

each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking

device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first computer program module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

27. (Currently Amended) A computer-readable medium for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the computer-readable medium carrying:

a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times;

a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first computer program module; and

one or more sequences of instructions, which, when executed by one or more processors, cause the one or more processors to carry out the steps of retrieving data from the mapping, and developing a second version number for a second computer program module of the one or more interacting computer program modules based on the mapping and based on a description of interface capabilities of the second module,

wherein compatibility is determined based on a first version number for the first computer program module and the second version number for the second computer program module.

28. (Currently Amended) A system for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the system comprising:

means for retrieving a stored mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module;

means for automatically [[developing]] assigning a second version number for a second computer program module of the one or more interacting computer program modules based on the mapping and based on a description of interface capabilities of the second module; and

means for determining compatibility based on a first version number for the first computer program module and the second version number for the computer program second module.

29. (Currently Amended) A computer system for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the system comprising:

a processor;

a computer-readable medium carrying

a stored mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module, and

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

retrieving data from the mapping;

[[developing]] automatically assigning a second version number for a second computer program module of the one or more interacting computer program modules based on the mapping and based on a description of interface capabilities of the second module; and

determining compatibility based on a first version number for the first computer program module and the second version number for the second computer program module.